**AIM: Practice and Build Factory Design Pattern**

**Code:** Example taken of Toys

|  |
| --- |
| import java.util.Scanner;  interface Toys{      void type(String toyType);  }  class Main {      public void type() {          System.out.println("Main Toy");      }      static void printMenu() {          System.out.println("1. Kid Toy");          System.out.println("2. Hard Toy");          System.out.println("3. Soft Toy");          System.out.println("Else Any Key Exit");      }      static int getChoice() {          Scanner sc = new Scanner(System.in);          int choice = sc.nextInt();          return choice;      }      public static void main(String[] args) {          int choice;          printMenu();          choice = getChoice();          while (choice > 0 && choice < 4) {              myToy toy;              switch (choice) {                  case 1:                      toy = new myToy();                      toy.type("Kid Toy");                      break;                  case 2:                      toy = new myToy();                      toy.type("Hard Toy");                      break;                  case 3:                      toy = new myToy();                      toy.type("Soft Toy");                      break;                  default:                      System.out.println("Exit");              }              printMenu();              choice = getChoice();          }      }  }  class myToy implements Toys {      String toyType;      public void type(String toyType) {          this.toyType = toyType;          System.out.println(toyType);      }      public String toString() {          return this.toyType;      }  } |

**Output:**

